**ITC LAB 14  
Arrays , CString , cin.getline (arrayname, size)**

**Q1.** Write a program in C++ to create two Cstring of size 20 each. Initialize it with your first name and last name respectively. Display the message “your first name and last starts with the same letter” if both starts with the same letter else “Both have different starting letter”

**Expected Output:  
Sample 1:**

Input your first name: asad  
Input your last name: nazir

“Both have different starting letter”

**Sample 2:**

Input your first name: atia  
Input your last name: Anees

your first name and last starts with the same letter A

**Q2.** Write a program in C++ to create a Cstring of size 20. Initialize it with your full names using cin.getline. Convert all the upper case letters into lower case (using + or - 32) if it exists else leave it as it is. Print the output as follows:

**Expected Output:**

Enter first name : sadia sAleem  
output: sadia saleem  
  
Enter first name : haider ali  
output: haider ali

**Q3.** Write a program in C++ to create two Cstring of size 20 each. Initialize it with two full names using cin.getline. Check if both names are same or not using strcmp. (Change every upper case to lower case if any before comparing). Display the message as follows :  
  
 **Expected Output:**

Enter first name : ahmed nazeer  
Enter second name: ahmed Nazeer  
Both names are same.   
  
Enter first name : babar ali  
Enter second name: asad liaqut  
Both names are different.

**Q4.** Write a program in C++ to create an integer array of size 10 with n elements each. (Where n>=1 and <=size). Input two indices from user in the range 0 to n-1 and swap the values at those indexes. Print the modified array after swapping as follows:

**Expected Output:**

Input n: 4  
Input array elements : 12 8 9 -5  
Input ind1: 2  
Input ind2: 0  
Modified array: 9 8 12 -5  
  
**Q5.** Write a program in C++ to create an integer array of size 10 with n elements each. (Where n>=1 and <=size). Find minimum value in the array and place it in the first location of array.  
Print the modified array after swapping as follows:

**Expected Output:**

Input n: 4  
Input array elements : 12 8 9 -5  
Modified array: -5 8 9 12

**Q6.** Write a program in C++ to create an integer array of size 10 with n elements each. (Where n>=1 and <=size). Sort the array in ascending order using above technique as discussed in classPrint the modified array after sorting as follows:

**Expected Output:**

Input n: 4  
Input array elements : 12 8 9 -5  
Sorted array: -5 8 9 12

**Q7.** Write a program in C++ to create an integer array of size 10 with n elements each. (Where n>=1 and <=size). Shift entire array one step to the right if and only if n<size. Display the shifted array as follows: ( Note : Reminding again that array location after being shifted can never be empty , as shifting is copy paste not cut paste)

**Expected Output:**

Input n: 4  
Input array elements : 12 8 9 -5  
Modified array: 12 12 8 9 -5

**Q8.** Write a program in C++ to create an integer array of size 10 with n elements each. (Where n>=1 and <=size).Input an index in the range 0 to n-1 and start your shifting an array one step to the right from that index if and only if n<size. Display the shifted array as follows: ( Note : Reminding again that array location after being shifted can never be empty , as shifting is copy paste not cut paste)

**Expected Output:**

Input n: 5  
Input array elements : 12 8 9 -5 7  
Input index from where to start shifting : 2  
Modified array: 12 8 9 9 -5 7

**Q9.** Write a program in C++ to create an integer array of size 10 with n elements each. (Where n>=1 and <=size).Input an index in the range 0 to n-1 and start your shifting an array one step to the right from that index if and only if n<size. After shifting place 0 at the shift index value. Display the shifted array as follows: ( Note : Reminding again that array location after being shifted can never be empty , as shifting is copy paste not cut paste)

**Expected Output:**

Input n: 5  
Input array elements : 12 8 9 -5 7  
Input index from where to start shifting : 2  
Modified array: 12 8 0 9 -5 7

**Q10.** Write a program in C++ to create two integer array of size 10 with n elements each. (Where n>=1 and <=size).Start your shifting an array one step to the left and keep the count while copying the lost element in the second array starting from 0 index to onwards till n becomes 0.  
Display the shifted array at each step and n as well as the second array as follows: ( Note : Reminding again that array location after being shifted can never be empty , as shifting is copy paste not cut paste).

**Expected Output:**

Input n: 5  
Input array elements: 12 8 9 -5 7  
Original array after shift # 1 : 8 9 -5 7  
Second array: 12   
count=1  
Updated n : 4  
Original array after shift #2 : 9 -5 7  
Second array: 12 8  
count=2  
Updated n : 3  
Original array after shift # 3 : -5 7  
Second array: 12 8 9  
count=3  
Updated n : 2  
Original array after shift # 4 : 7  
Second array: 12 8 9 -5  
count=4  
Updated n : 1  
Original array after shift # 5:   
Second array: 12 8 9 -5 7  
count=5  
Updated n : 0

**Q11.** Write a program in C++ to create a Cstrings of size 60. Initialize it with a sentence of your choice using strcpy as discussed in class. (Note a sentence consist of multiple words with spaces in between and ends at full stop). Count total words, total capital letters, total small letters, total symbols (excluding space), total digits in the sentence and display all the count as follows:

**Expected Output:**

**Sample 1:**

Your stored sentence is: Where there is a will there is a way.  
Total words = 9  
Total capital letters = 1  
Total small letters = 27  
Total symbols = 1  
Total digits = 0

**Sample 2:**

Your stored sentence is: ASSIGNMENT #7 submission has been created.

Total words = 6  
Total capital letters = 10  
Total small letters = 24  
Total symbols = 2  
Total digits = 1

**Q12.** Write a program in C++ to create a Cstrings of size 60. Initialize it with a sentence of your choice using strcpy as discussed in class. (Note a sentence consist of multiple words with spaces in between and ends at full stop). Count all the occurrences of a character n input by user in it and display and display the count as follows:

**Expected Output:  
Sample 1:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: e  
e occurs 6 times

**Sample 2:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: z  
z occurs 0 times

**Sample 3:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: $  
$ occurs 0 times

**Q13.** Write a program in C++ to create a Cstring of size 60. Initialize it with a sentence of your choice using strcpy as discussed in class. (Note a sentence consist of multiple words with spaces in between and ends at full stop). Display all the indexes where the character n input by user occur in that cstring as follows:

**Expected Output:  
Sample 1:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: e  
e occurs at 2,4,8,10,24,26 indexes in array

**Sample 2:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: z  
z doesn’t exist in array

**Sample 3:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: $  
$ doesn’t exist in array

**Q14.** Write a program in C++ to create a Cstring of size 60. Initialize it with a sentence of your choice using strcpy as discussed in class. (Note a sentence consist of multiple words with spaces in between and ends at full stop). Store all the indexes of the occurrences of a character n input by user in another integer array name out\_array of size 60 and display all those stored indexes from array as follows:

**Expected Output:  
Sample 1:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: e

e occurs 6 times at the following indexes stored in out\_array as follows:  
out\_array[0] = 2  
out\_array[1] = 4  
out\_array[2] = 8  
out\_array[3] = 10  
out\_array[4] = 24  
out\_array[5] = 26

**Sample 2:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: z  
z doesn’t exist in array

**Sample 3:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: $  
$ doesn’t exist in array

**Q15.** Write a program in C++ to create a Cstring of size 60. Initialize it with a sentence of your choice using strcpy as discussed in class. (Note a sentence consist of multiple words with spaces in between and ends at full stop). Store all the indexes of the occurrences of a character n input by user in another integer array name out\_array of size 60 . Replace the second occurrence with # sign display output as follows:

**Expected Output:  
Sample 1:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: e

e occurs 6 times at the following indexes stored in out\_array as follows:  
out\_array[0] = 2  
out\_array[1] = 4  
out\_array[2] = 8  
out\_array[3] = 10  
out\_array[4] = 24  
out\_array[5] = 26

Your modified sentence is : Wher# there is a will there is a way.

**Sample 2:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: y  
y occurs 1 times at the following indexes stored in out\_array as follows:  
out\_array[0] = 35

**Sample 3:**

Your stored sentence is: Where there is a will there is a way.  
Input character to count occurrences: $  
$ doesn’t exist in array